

LISTING OF CLAIMS

Claim 1 (Currently Amended): A method of enabling a user to extend a work flow for synchronization/consolidation of data between at least two data sources, said work flow for execution in a meta directory server, said method comprising:

5 providing, by a designer, a sequence of built-in tasks which together when executed implement said work flow for synchronization/consolidation of data between at least two data sources, a built-in task in said sequence of built-in tasks containing an extension point at a point of interest in said work flow for users,

10 said work flow being designed for execution by multiple users as corresponding instances,

wherein each user is provided the ability to specify a corresponding custom task associated with said extension point without editing said work flow, the custom task specified by a user containing corresponding program logic to provide a customization desired by the user,

15 said flow being designed to execute the specified desired custom task in the corresponding instance if specified by corresponding user at said extension point;

receiving from said user data indicating a custom task associated with said extension point, wherein said custom task is separate from said sequence of built-in tasks and contains a program logic specified by said user for the corresponding desired customization;

20 executing said custom task when said extension point is reached during execution of said built-in task in an instance of said work-flow for said user; and

25 continuing execution of said sequence of built-in tasks from said extension point in said built-in task after executing said custom task in said instance such that all of said sequence of built-in tasks are executed to complete synchronizing/ consolidating data between said two data sources.

Claim 2 (Canceled)

Claim 3 (Original): The method of claim 1, wherein said custom task contains an another extension point, said method further comprises receiving from said user data indicating 30 an another custom task to be executed when said another extension point is reached during execution of said custom task.

Claim 4 (Previously Presented): The method of claim 3, further comprising:

determining a corresponding set of extension points available in each of said sequence of built-in tasks;

displaying each of said set of extension points associated with a corresponding one of said sequence of built-in tasks;

5 displaying said custom task and said another custom task; and

enabling said user to specify said custom task associated with said extension point, and said another custom task associated with said another extension point.

10 Claim 5 (Previously Presented): The method of claim 3, further comprising enabling said user to specify that said custom task is to be executed synchronously, wherein execution of said sequence of built-in tasks is suspended at said extension point during execution of said custom task, and

wherein execution o said sequence of built-in tasks is resumed after completion of execution of said custom task such that said custom task is executed in a synchronous manner.

15

Claim 6 (Previously Presented): The method of claim 3, further comprising enabling said user to specify that said custom task is to be executed asynchronously, wherein said custom task is executed in parallel with execution of built-in task from said extension point such that said custom task is executed in an asynchronous manner.

20

Claim 7 (Canceled)

Claim 8 (Previously Presented): The method of claim 1, wherein at least one of said two data sources comprises a relational database.

25

Claim 9 (Original): The method of claim 3, further comprising providing an utility to indicate that a specific one of said extension points is reached.

30 Claim 10 (Previously Presented): The method of claim 3, further comprising providing an utility in each of said sequence of built-in tasks and said custom task, wherein said utility indicates extension points available in a corresponding task.

Claim 11 (Currently Amended): A computer readable medium storing one or more

sequences of instructions for causing a meta directory server to enable a user to extend a work flow for synchronization/consolidation of data between at least two data sources, wherein execution of said one or more sequences of instructions by one or more processors contained in said meta directory server causes said meta directory server to perform the actions of:

5 providing, by a designer, a sequence of built-in tasks which together when executed implement said work flow for synchronization/consolidation of data between at least two data sources, a built-in task in said sequence of built-in tasks containing an extension point at a point of interest in said work flow for users,

10 said work flow being designed for execution by multiple users as corresponding instances,

wherein each user is provided the ability to specify a corresponding custom task associated with said extension point without editing said work flow, the custom task specified by a user containing corresponding program logic to provide a customization desired by the user,

15 said flow being designed to execute the specified desired custom task in the corresponding instance if specified by corresponding user at said extension point;

 receiving from said user data indicating a custom task associated with said extension point, wherein said custom task is separate from said sequence of built-in tasks and contains a program logic specified by said user;

20 executing said custom task when said extension point is reached during execution of said built-in task in an instance of said work-flow for said user; and

 continuing execution of said sequence of built-in tasks from said extension point in said built-in task after executing said custom task in said instance such that all of said sequence of built-in tasks are executed to complete synchronizing/ consolidating data between said two data sources.

25

Claim 12 (Canceled):

30 Claim 13 (Original): The meta directory server of claim 11, wherein said custom task contains an another extension point, further comprises receiving from said user data indicating an another custom task to be executed when said another extension point is reached during execution of said custom task.

Claim 14 (Previously Presented): The meta directory server of claim 13, further

comprising:

determining a corresponding set of extension points available in each of said sequence of built-in tasks;

5 displaying each of said set of extension points associated with a corresponding one of said sequence of built-in tasks;

displaying said custom task and said another custom task; and

enabling said user to specify said custom task associated with said extension point, and said another custom task associated with said another extension point.

10 Claim 15 (Previously Presented): The meta directory server of claim 13, further comprising enabling said user to specify that said custom task is to be executed synchronously, wherein execution of said sequence of built-in tasks is suspended at said extension point during execution of said custom task, and

15 wherein execution o said sequence of built-in tasks is resumed after completion of execution of said custom task such that said custom task is executed in a synchronous manner.

Claim 16 (Previously Presented): The meta directory server of claim 13, further comprising enabling said user to specify that said custom task is to be executed asynchronously, wherein said custom task is executed in parallel with execution of built-in task from said 20 extension point such that said custom task is executed in an asynchronous manner.

Claim 17 (Canceled)

Claim 18 (Previously Presented): The meta directory server of claim 11, wherein at least 25 one of said two data sources comprises a relational database.

Claim 19 (Original): The meta directory server of claim 13, further comprising providing an utility to indicate that a specific one of said extension points is reached.

30 Claim 20 (Previously Presented): The meta directory server of claim 13, further comprising providing an utility in each of said sequence of built-in tasks and said custom task, wherein said utility indicates extension points available in a corresponding task.

Claim 21 (Currently Amended): A meta directory server enabling a user to extend a work flow for synchronization/consolidation of data between at least two data sources, said meta directory server comprising:

means for providing, by a designer, a sequence of built-in tasks which together when executed implement said work flow for synchronization/consolidation of data between at least two data sources, a built-in task in said sequence of built-in tasks containing an extension point at a point of interest in said work flow for users,

said work flow being designed for execution by multiple users as corresponding instances,

wherein each user is provided the ability to specify a corresponding custom task associated with said extension point without editing said work flow, the custom task specified by a user containing corresponding program logic to provide a customization desired by the user,

said flow being designed to execute the specified desired custom task in the corresponding instance if specified by corresponding user at said extension point;

means for receiving from said user data indicating a custom task associated with said extension point, wherein said custom task is separate from said sequence of built-in tasks and contains a program logic specified by said user for the corresponding desired customization;

means for executing said custom task when said extension point is reached during execution of said built-in task; and

means for continuing execution of said sequence of built-in tasks from said extension point in said built-in task after executing said custom task in said instance such that all of said sequence of built-in tasks are executed to complete synchronizing/ consolidating data between said two data sources.

25 Claim 22 (Canceled):

Claim 23 (Original): The meta directory server of claim 21, wherein said custom task contains an another extension point, further comprises means for receiving from said user data indicating an another custom task to be executed when said another extension point is reached during execution of said custom task.

Claim 24 (Previously Presented): The meta directory server of claim 23, further comprising:

means for determining a corresponding set of extension points available in each of said sequence of built-in tasks;

means for displaying each of said set of extension points associated with a corresponding one of said sequence of built-in tasks;

5 means for displaying said custom task and said another custom task; and

means for enabling said user to specify said custom task associated with said extension point, and said another custom task associated with said another extension point.

10 Claim 25 (Previously Presented): The meta directory server of claim 23, further comprising enabling said user to specify that said custom task is to be executed synchronously, wherein execution of said sequence of built-in tasks is suspended at said extension point during execution of said custom task, and

wherein execution o said sequence of built-in tasks is resumed after completion of execution of said custom task such that said custom task is executed in a synchronous manner.

15

Claim 26 (Previously Presented): The meta directory server of claim 23, further comprising enabling said user to specify that said custom task is to be executed asynchronously, wherein said custom task is executed in parallel with execution of built-in task from said extension point such that said custom task is executed in an asynchronous manner.

20

Claim 27 (Canceled)

Claim 28 (Previously Presented): The meta directory server of claim 21, wherein at least one of said two data sources comprises a relational database.

25

Claim 29 (Original): The meta directory server of claim 23, further comprising an utility means to indicate that a specific one of said extension points is reached.

30 Claim 30 (Previously Presented): The meta directory server of claim 23, further comprising an utility means in each of said sequence of built-in tasks and said custom task, wherein said utility means indicates extension points available in a corresponding task.

Claim 31 (Currently Amended): A meta directory server enabling a user to extend a

work flow for synchronization/consolidation of data between at least two data sources, said meta directory server comprising:

a task registry block storing data related to a sequence of built-in tasks which together when executed implement said work flow for synchronization/consolidation of data between at 5 least two data sources, a built-in task in said sequence of built-in tasks containing an extension point at a point of interest in said work flow for users,

said work flow being designed for execution by multiple users as corresponding instances,

wherein each user is provided the ability to specify a corresponding custom task 10 associated with said extension point without editing said work flow, the custom task specified by a user containing corresponding program logic to provide a customization desired by the user,

said flow being designed to execute the specified desired custom task in the corresponding instance if specified by corresponding user at said extension point;

15 a user interface module receiving from said user, data indicating a custom task associated with said extension point, wherein said custom task is separate from said sequence of built-in tasks and contains a program logic specified by said user for the corresponding desired customization; and

20 work-flow manager module for executing said custom task when said extension point is reached during execution of said built-in task, and in addition for continuing execution of said sequence of built-in tasks from said extension point in said built-in task after executing said custom task in said instance such that all of said sequence of built-in tasks are executed to complete synchronizing/ consolidating data between said two data sources.

Claim 32 (Canceled):

25

Claim 33 (Original): The meta directory server of claim 31, wherein said custom task contains an another extension point, wherein said user interface further receives data indicating an another custom task to be executed when said another extension point is reached during execution of said custom task.

30

Claim 34 (Previously Presented): The meta directory server of claim 33, wherein said user interface modules displays each of said set of extension points associated with a corresponding one of said sequence of built-in tasks, and enables said user to specify said

custom task associated with said extension point and said another custom task associated with said another extension point.

Claim 35 (Previously Presented): The meta directory server of claim 33, wherein said user interface enables said user to specify that said custom task is to be executed synchronously, wherein execution of said sequence of built-in tasks is suspended at said extension point during execution of said custom task, and

wherein execution of said sequence of built-in tasks is resumed after completion of execution of said custom task such that said custom task is executed in a synchronous manner.

10

Claim 36 (Previously Presented): The meta directory server of claim 33, wherein said user interface enables said user to specify that said custom task is to be executed asynchronously, wherein said custom task is executed in parallel with execution of built-in task from said extension point such that said custom task is executed in an asynchronous manner.

15